Journal of Physics: Conference Series **513** (2014) 042017 doi:10.1088/1742-6596/513/4/042017

Disk storage at CERN: Handling LHC data and beyond

X. Espinal, G. Adde, B. Chan, J. Iven, G. Lo Presti, M. Lamanna, L. Mascetti, A. Pace, A. Peters, S. Ponce, E. Sindrilaru

CERN European Laboratory for Particle Physics 1211 Genève (Switzerland)

E-mail: xavier.espinal@cern.ch

Abstract. The CERN-IT Data Storage and Services (DSS) group stores and provides access to data coming from the LHC and other physics experiments. We implement specialised storage services to provide tools for optimal data management, based on the evolution of data volumes, the available technologies and the observed experiment and users' usage patterns. Our current solutions are CASTOR, for highly-reliable tape-backed storage for heavy-duty Tier-0 workflows, and EOS, for disk-only storage for full-scale analysis activities. CASTOR is evolving towards a simplified disk layer in front of the tape robotics, focusing on recording the primary data from the detectors. EOS is now a well-established storage service used intensively by the four big LHC experiments. Its conceptual design based on multi-replica and in-memory namespace, makes it the perfect system for data intensive workflows. The LHC-Long Shutdown 1 (LS1) presents a window of opportunity to shape up both of our storage services and validate against the ongoing analysis activity in order to successfully face the new LHC data taking period in 2015. In this paper, the current state and foreseen evolutions of CASTOR and EOS will be presented together with a study about the reliability of our systems.

1. Introduction

The data Storage and services group at CERN provides many services to the physics community. Here we describe two: CASTOR and EOS which are essential for LHC data taking, data processing, data distribution and analysis.

2. Large Scale Storage Systems at CERN: CASTOR and EOS

CASTOR is the Hierarchical Storage Management System for handling disk and tape layers. Born in 1999, it holds 92PB of data and 350M files in a common namespace. The current deployment status consists of 6 production instances: one for each major LHC experiment, one for the rest of the users and one dedicated to tape repacking activities. CASTOR fully relies on an ORACLE database backend which runs the namespace and handles the head nodes, disk servers and tape server logics. CASTOR's main role consists of data recording activities and data export and import. CASTOR's native protocol is rfio but it also supports xroot and gridftp. The current disk layout configuration is RAID-1. Access is secured via kerberos and X509.

During LS1 the CASTOR software evolution is concentrating on its main functionality (data recording) and some obsolete features are being removed (ie. file updates, legacy protocols). Code is being simplified and optimized and part of the logics moved from C++ code to database PL/SQL procedures in the Data Base to run closer to the data. These efforts are reflected in the 1kHz rates that CASTOR can handle as when compared to the O(50)Hz limitation we had some years ago (old scheduling system) [1, 2].

EOS started its production phase in 2011 and currently holds 20PB of data and 158M files. It is a disk-only storage solution mainly focused on analysis and fast data processing with a very low access latency (ms to s) thanks to the multi-replication across nodes and JBOD disk layout. Fast metadata access is guaranteed by the in-memory resident per instance namespace. Xroot is the principal access protocol. Protocols such as gridftp, fuse mount and http are also supported. Authentication is done by Kerberos/X509. The four big LHC experiments are using EOS, and a shared instance for non-LHC experiments started its production phase recently [3] [4].

The Data Storage and Services group at CERN is offering to the experiments at CERN these two large scale storage systems that can be combined to meet the HEP experiments requirements of today.

3. Disk and tape separated infrastructure

HEP storage use patterns are changing and we have witnessed a shift from HSM to infrastructures where the experiment (sometimes the user) keep control of the data placement. Different storage types are already handled by the experiments as if they were de facto separated. They successfully handled prestaging, pinning and data migration during the first LHC run. On the other hand we observed that shared disk/tape infrastructures potentially leads to inefficiencies in exploiting tape infrastructures performance when under load as the disk competition between read and writing activity have an impact on the remaining throughput capacity left on the disk server to feed tape drive. These two ingredients are the recipe for a next generation HSM with different storage types, and the evolution of our storage systems is moving in this direction. EOS is the disk endpoint designed for high concurrency and low latency while keeping data reliability very high. CASTOR is a long-term storage endpoint oriented to boost tape infrastructure performance and to have a solid workflow from disk to tape and vice versa. CASTOR is protected for data recording activities with limited number of transfer slots and IP prioritisation for tape recording [5].
We envision two main different and co-existing scenarios to be considered. The crucial point is where to stream the raw data and this is chosen by the experiments:

- Raw data is sent to CASTOR and then copied into EOS for analysis and production activities. Experiment computing resources at the pits is being increased and there will be less need to run post-DAQ data processing previous to data consolidation. This option envisions all data analysis and processing on EOS and keeps CASTOR as a data back-up unit or use it optionally as a buffer for Tier-1 data export/import.

- Raw data is sent to EOS and share the resources with the following activities: data analysis, data processing activities and the WLCG data import/export. CASTOR in this case is used as an asynchronous high latency endpoint for data backup, effectively being a cold storage endpoint. The benefit is that the resultant infrastructure in EOS will be larger and hence able to deliver more IO by offering more transfer slots. This approach would imply an intensive and realistic testing during the upcoming year.

4. EOS system split: CERN-Wigner

The new CERN-IT datacenter at the Wigner institute (Budapest) is entering the production phase. The EOS storage service will be split between our two datacenters as the system demonstrated a good WAN performance and also due to the volume of disk servers. This brings interesting challenges as running a distributed system implies awareness of data locality. Geo-tagging is added at the file metadata level and at client level so the goal is to minimise network traffic between CERN and Wigner. For the production and analysis jobs running on batch systems the goal is for jobs to get the nearest possible replica.

However, at data placement time we want to maximise the distance between the replicas to increase data reliability. Metadata access is being designed to minimise RTTs between client and namespace queries by running a slave NS server at the remote datacenter (read-only) so that clients can contact the closest headnode and fallback in case of failure/incident. This opens the window to geo-redundancy by having a master/slave namespace configuration between both computing centres (detailed information on [6]).

5. Disk layouts: introducing RAIN

Maximising IO throughputs and data reliability are the prerequisites for choosing the correct disk layout. RAID-1 will continue to be used for CASTOR as this aligns well with the foreseen evolution of CASTOR as the tape endpoint. That said, there is no software limitation to implement JBOD or double-replication on CASTOR if needed.

On EOS, it is possible to have different disk layouts to increase IO throughputs and disk space savings while keeping the data reliability high by striping files across nodes via a Redundant Array of Independent Nodes (RAIN). The advantages are high scalability and high reliability. The drawbacks are the required computational effort (as a consequence of the non-sequential writing file is re-composed on the client server) and the increase of IOPS between the client and the servers as all the communication is done over the network [9].

There are currently two possible layouts for RAIN:

- *a) RAID-DP* made up of 4 data stripes and 2 parity stripes that show-up to be very fast as the parity computation uses XOR operations [7].
- *b) ReedSolomon* which uses polynomials and matrix transformations for error correction (Jerasure library). The encoding/decoding is more computationaly intensive when compared with RAID-DP. On the other hand it is more flexible than RAID-DP as it supports various combinations of data and parity stripes. An added value is that the reliability is independent of the chunk size and can be easily tuned by increasing the number of stripes.

Client reads in RAIN use parallel IO configuration where the file is collected and recreated by the client machine hence allowing all the disk servers to stream data chunks in parallel. This mode use the new XrootD client (XrdCl) and the communication is fully parallel and asynchronous (more information on [9]).

6. Agile Storage

CERN-IT is changing the Installation and Configuration Management Systems (CMS). The old infrastructure based on Quattor is being replaced by the new AGILE infrastructure based on commercial software: Puppet, Foreman, Hiera and home-brewed tools for the installation and Configuration Management System; mcollective for the cluster orchestration; git for the Version Control System. There are many levels on which this affects the normal operations workflows. Full code was rewritten for the different Configuration Items, operational tools are being adapted and the mechanism for host installation and host status management are being put in place. In 2012 the IT-DSS group pioneered the first services in production under the Agile infrastructure and has been following the project evolution since its early phase. The storage services in production has evolved up together with Agile project maturity. Today there is a full CASTOR instance in production along with a dedicated experiment diskpool. EOS also started running some disk servers in production under the new infrastructure (further information about the AGILE infrastructure on [10]).

Reason	Percentage (%)
Disk	77%
Power Supply	5%
Memory	5%
Motherboard	4%
RAID controller	2.5%
IPMI	2%
BBU	2%

Reason	Percentage (%)
Cooling System/Fans	1%
Backplane	0.8%
Internal cabling	0.7%
CPU	0.6%
Unknown	0.4%
Serial card	0.1%
Network card	0.1%

Table 1. Disk server failures statistics collected at CERN computing centre between 2009 and 2013

7. Storage Systems Reliability

Data loss in large scale storage systems is an unavoidable fact. Hardware failures, software bugs, silent corruption and operational (human) errors contribute to the data loss chain. These, together with today's data *explosion* leads to the massive storage installations being affected by the smallest failure probabilities. Hence design of storage systems need to take this into account. At CERN the Annual Failure Rate (AFR) observed during the last years for hard drives is approx. 2.5% and approx. 1% for raid controllers, both of which vary between different vendors (source [11] and [12, 13, 14]).
File lost rate on tapes is very low but visible for systems with hundreds of millions of files (10^8). Data on tape is cold so data corruption is spotted during repack activities and scrubbing runs which happen some time after the data is recorded.
On the disk side, double copy on different nodes with JBOD configuration systems (EOS) are very resilient to hardware failures as the files are distributed across different nodes. On the other hand, double copy single disk mirroring (CASTOR with RAID-1) configurations are heavily impacted by double disk failures unless used as a cache in front of tape infrastructure.

The goal of this section is to know the design reliability of our large scale storage systems from the disk failure perspective by taking a simple approach: treat our systems as dual parallel systems with a known disk failure rate as the input parameter and then estimate the impact of data loss during the system recovery time. Recovery time is defined as the period of time whilst the system is running in degraded mode but still able to repair itself: this means replica number adjustment in EOS and tape migration (or disk rebuild) in CASTOR.

7.1. Hardware failures

During last years we have collected statistics for the hardware failures observed in the computing centre. About 3000 disk servers have been running since 2009. The statistics for broken parts are shown in Table 1. There are basically two types of failures that can lead to data loss, raid controllers and disk failures. Other types of failures can lead to an effective file truncation but the impact is very low as the client reports a failed transfer. RAID controller failures has not been considered when computing the theoretical system reliabilities as the impact when compared with disk failures is low: the raid controller repair success is $> 95\%$ and data could be retrieved from tape most of the times.

The data loss statistics gathered during the last two years for CASTOR and EOS are shown in Table 2. The large values for CASTOR lost files during 2011 and 2012 are a result of several RAID controller failures involving disk-only data. The trend of lost files is decreasing as an effect of EOS maturity and CASTOR improvements together with the decommissioning of disk-only pools.

Dataloss	2011 (%)	2012 (%)	2013 (%)
CASTOR	113353 (10^{-4})	12711 (10^{-5})	43 (10^{-7})
EOS	133 (10^{-5})	182 (10^{-6})	22 (10^{-8})

Table 2. Files loss stats in number of files per year and percentage over namespace entries. Large values for CASTOR lost files during 2011 and 2012 are a result of several RAID controller failures involving D1T0 data.

7.2. System reliability estimation

System reliability is evaluated in this section taking into account the different scenarios to data loss from the disk failures perspective without taking into account the systematic effects mentioned previously. The different scenarios are:

- *CASTOR:* A file is transferred to disk and the migration job process is scheduled at the Data Base. Before the migration to tape is executed the file is on two disks (RAID-1). A tape mount starts based on several triggers with the dominant one being the time. Having a double disk failure within this time window leads to data loss.

- *EOS:* Consider a disk failure. Having a second disk failure before adjustment of the number of replicas leads to data loss. During replica adjustment time all disks belonging to the instance (or group inside the instance) stream data among themselves to re-replicate files affected by the first disk failure (minus one on the replica factor). The number of affected files depends on the overlap of data among disks and this depends on the number of files, disk size and number of disks.

To estimate the reliability we consider our storage systems as dual parallel systems [15, 16] where each subcomponent is a disk that follows an exponential failure distribution. We took the assumption that each subcomponent (disk) is considered as an independent system, we obviate possible disk failure correlations as they are difficult to quantify: on EOS the possible correlations are minimal as the disks sharing data are distributed across nodes, while on CASTOR disks holding the same data lie on the same diskserver and the effect might be sizeable in case of incidents (power outage, disk switching, etc). Hence the reliability of the systems can be defined as follows, where T is the system recovery time after a first disk failure occurs. The system recovery time is the time taken by the system to compensate the disk failure (raid rebuild in the case of CASTOR and copy re-replication in EOS):

$$R(t) = 1 - (1 - e^{-\lambda_1 t}) \cdot (1 - e^{-\lambda_2 t}) \tag{1}$$

The system hazard rate λ_s can be derived from (eq. 1):

$$\lambda_s(t) = \frac{\text{Density Function}}{\text{Survival Function}} = \frac{-dR(t)/dt}{R(t)} \tag{2}$$

$$\lambda_s(t) = 2\lambda \frac{1 - e^{-\lambda t}}{2 - e^{-\lambda t}} \tag{3}$$

To evaluate the impact on lost files one should consider the different data loss scenarios mentioned before and parametrize them. On EOS this is the mentioned overlap factor that can be approximated by the number of files per disk divided by the number of disks. Using conventional storage parameters this translates into (where d_s is disk size , $\overline{f_s}$ is the mean file size and C_T is total capacity installed):

$$Overlap = \frac{NFilesDisk}{Ndisks} = \frac{d_s/\overline{f_s}}{C_T/d_s} = \frac{d_s^2}{C_T \cdot \overline{f_s}} \tag{4}$$

The impact in case of double failure is:

$$Loss = 2\lambda \frac{1 - e^{-\lambda T}}{2 - e^{-\lambda T}} \cdot \frac{d_s^2}{C_T \cdot \overline{f_s}} \tag{5}$$

In CASTOR the overlap parameter is the number of files accumulated on the disk during the waiting time for tape migration (no more disk-only files). Thus the overlap parameter is simply the rate of files per disk which we estimated based on normal operation values during LHC data taking: 1PB/week which is 10TB/h. So the overlap in this case is:

$$Overlap(t) = \frac{accumulatedFiles}{disk} = \frac{fileRate \cdot T}{Ndisks} = \frac{10(TB/h)/\overline{f_s}}{C_T/d_s} = 10 \cdot \frac{d_s}{C_T \cdot \overline{f_s}} \cdot T \tag{6}$$

And the impact in case of double failure is:

$$Loss = 2\lambda \frac{1 - e^{-\lambda T}}{2 - e^{-\lambda T}} \cdot 10 \cdot \frac{d_s}{C_T \cdot \overline{f_s}} \cdot T \tag{7}$$

We consider the systems either repaired or broken after 24h as at this time manual intervention is triggered through alarms (expired draining or old files pending to be migrated). The 24h corresponds to a slow re-replication in EOS or long waiting tape migration time for CASTOR (usually all data is sent to tape after 4h) so both are worst case scenarios. Having fixed the maximum system repair time it is possible to estimate the impact of the different parameters on the system reliability: disk sizes, total capacity, network-disk-tape speeds, etc. One thing to take into account is the definition of our systems when estimating the reliabilities. In EOS the system is defined by all the nodes composing the instance (or a subgroup of filesystems within it). In CASTOR the system is a single server where the effect of having more capacity (servers) is reflected in the dispersion among the disks of the incoming file rate (files pending to be migrated to tape), hence the more the servers the less the file rate per disk.

To illustrate with an example we have defined a toy system consisting of 1PB and a mean file size of 1GB to see the impact of the disk size and the elapsed time before the system is fixed Figure 1. We considered a conservative Hourly Failure Rate for disks of $\lambda = 5 \cdot 10^{-6}$ (corresponding to AFR=5%). The file lost impact for the systems considering generic 2TB disks and considering the system broke after the maximum recovery time of 24h are: 10^{-8} for EOS and 10^{-7} for CASTOR. High reliability can be achieved on both systems.

8. Summary

Our storage systems successfully catered with LHC and non-LHC experiment needs during the last years. We continued evolving them during the LS1 to face the upcoming Run2 challenge. The actual deployment status and the work in progress of our Large Scale Storage Systems in production have been described: disk and tape separation for storage; disk layouts to improve reliability and streaming capacity; geographically distributed storage systems; migration to a new configuration management system. Finally the system reliability estimations for CASTOR and EOS have shown that dual parallel systems of different types can be highly reliable when moving towards specialisation.

References

[1] G. Lo Presti et al. "CASTOR: A Distributed Storage Resource Facility for High Performance Data Processing at CERN," IEEE Conference on Mass Storage Systems and Technologies 2007.

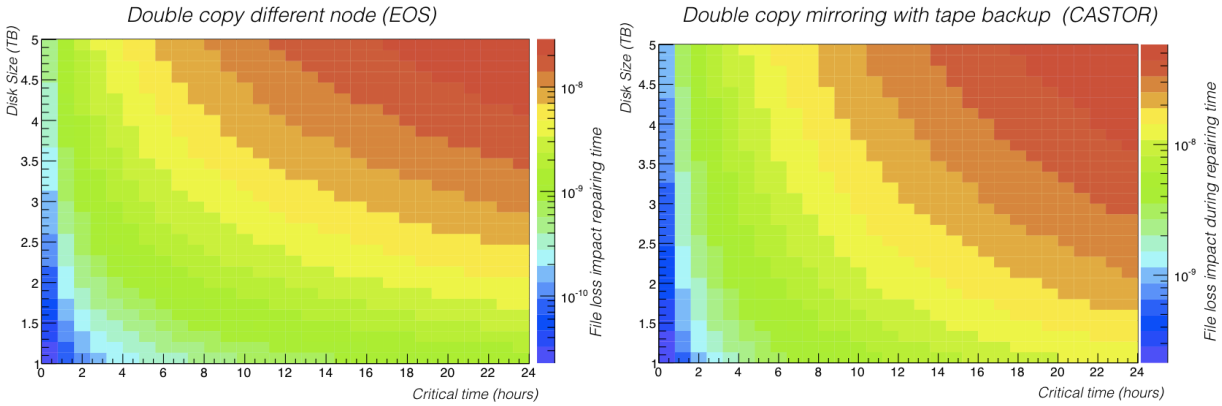

Figure 1. Lost files impact as a function of disk size and time to reparation. Left: EOS Right: CASTOR for a model with 1PB and mean file size of 1GB (1M files)

[2] G. Lo Presti et al. "Streamlining CASTOR to manage the LHC data torrent" CHEP 2013

[3] A. J. Peters and L. Janyst, "Exabyte scale storage at CERN"

[4] ROOT, an Object-Oriented Data Analysis Framework. http://root.cern.ch.

[5] E. Cano "System level traffic shaping in disk servers with heterogeneous protocols" CHEP 2013

[6] J. Iven et al. "di-EOS - "distributed EOS": Initial experience with split-site persistency in a production service" CHEP 2013

[7] RAID-DP Network Appliance, Inc. TR-3298 [12/2006], http://tarasistem.com/docs/NetApp/raiddp.pdf

[8] S. B. Wicker and V. K. Bhargava "Reed-Solomon Codes and Their Applications" IEEE Press, Piscataway, NJ, 1994

[9] A. J. Peters, E. A. Sindrilaru and P. Zigann "Evaluation of software based redundancy algorithms for the EOS storage system at CERN" J. Phys. Conf. Ser. **396** (2012) 042046.

[10] P. Andrade et al. "Review of CERN Data Centre Infrastructure" J. Phys. Conf. Ser. **042002** (2012)

[11] Numbers extracted from statistics collected at CERN CC. Private communications with CERN-IT Procurement Team members.

[12] M. C. dos Santos, D. Waldron "Observations made while running a multi-petabyte storage system" Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium

[13] E.Pinheiro, W. D. Weber and L. A. Barroso "Failure Trends in Large Disk Drive Population" Proceedings of the 5th USENIX Conference on File and Storage Technologies, February 2007

[14] B. Schroeder, G. A. Gibson "Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You?" Proceedings of the 5th USENIX Conference on File and Storage Technologies, February 2007

[15] Hoyland, A. and M. Rausand "System Reliability Theory: Models and Statistical Methods" Wiley, NY, 1994 ISBN 0-471-47133-X

[16] J.L. Romeu "Understanding Series and Parallel Systems Reliability" START 2004-5 S&P SYSREL (Vol.11, N.5)